

Contenido

INTRODUCCIÓN	1
Tipo de dependencias.....	3
Formas Normales.	5
1.ª forma normal	5
Dominios Estructurados:.....	5
Propietarios de multimonedas	5
2ª Forma Normal.....	7
3ª Forma Normal	10

INTRODUCCIÓN

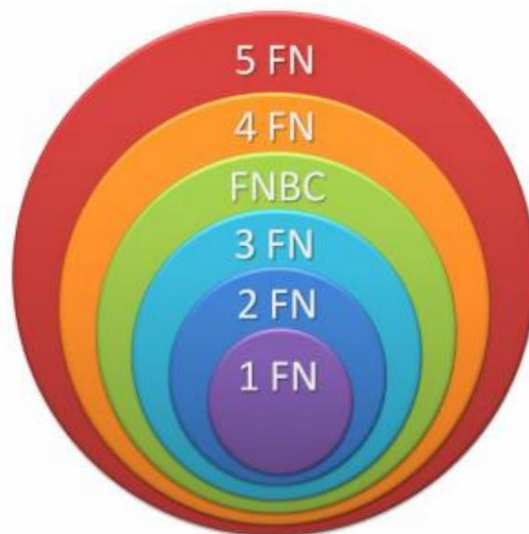
Normalización: Proceso que consiste en imponer en las tablas del modelo Lógico Relacional una serie de restricciones a través de un conjunto de transformaciones consecutivas. Este proceso garantizará que las tablas contienen los atributos necesarios y suficientes para describir la realidad de la entidad que representan, permitiendo separar aquellos atributos que por su contenido podrían generar la creación de otra mesa.

A principios de la década de los setenta, concretamente en 1972, Edgar Frank Codd (https://es.wikipedia.org/wiki/Edgar_Frank_Codd) establece una técnica para llevar a cabo el diseño de la estructura lógica de los datos representados a través del modelo relacional, en la que denominó normalización. Pero esta técnica no debe utilizarse para el diseño de la base de datos, sino como un proceso de refinamiento que debe aplicarse después de lo que conocemos como “paso a tablas”, o lo que formalmente se denomina traducción del esquema conceptual a el esquema lógico. Este proceso de refinamiento conseguirá los siguientes objetivos:

- Minimizar la redundancia de los datos: repetición de datos en un sistema.
- Suprimir dependencias erróneas entre atributos.
- Optimizar los procesos de inserción, modificación y borrado en la base de datos.

En la práctica, si la BD se ha diseñado haciendo uso de modelos semánticos como el modelo E/R, no suele ser necesaria la normalización. Por otra parte si nos proporcionan una base de datos creada sin realizar un diseño previo, es muy probable que necesitemos normalizar.

En la teoría de bases de datos relacionales, las formas normales (FN) proporcionan los criterios para a determinar el grado de vulnerabilidad de una mesa a inconsistencias y anomalías lógicas. Cómo más alta sea la forma normal aplicable a una mesa, menos vulnerable será a inconsistencias y anomalías. Edgar F. Codd originalmente definió las tres primeras formas normales (1FN, 2FN, y 3FN) en 1970. Estas formas normales se han resumido como requiriendo que todos los atributos sean atómicos, dependan de la clave completa y en forma directa (no transitiva). La forma normal de Boyce-Codd (FNBC) fue introducida en 1974 por los dos autores que aparecen en la su denominación. Las cuarta y quinta formas normales (4FN y 5FN) se ocupan específicamente de la representación de las relaciones muchos a muchos y uno a muchos entre los atributos y fueron introducidas por Fagin en 1977 y 1979 respectivamente. Cada forma normal incluye a las anteriores.



El proceso de normalización se basa en el análisis de dependencias entre atributos. Para ello tendrá en cuenta los conceptos de: dependencia funcional, dependencia funcional completa y dependencia transitiva.

Es un proceso que se realiza en distintas etapas secuenciales. Cada etapa está asociada a una forma normal, que establece unos requisitos a cumplir por la mesa sobre la que se aplica. Existen diversas formas normales: Primera, Segunda, Tercera, Boyce-Codd, Cuarta, Quinta y Dominio-Clave. Como hemos indicado, el paso de una forma normal a otra es consecutivo, si no se satisface una determinada forma normal no puede pasarse al análisis de la siguiente. Según vamos avanzando en la normalización, los requisitos a cumplir serán cada vez más restrictivos, lo que hará que

nuestro esquema relacional sea cada vez más robusto. Como norma general, para garantizar que no existan problemas en la actualización de datos, es recomendable aplicar el proceso de normalización hasta Tercera Forma Normal o incluso hasta Forma Normal de Boyce-Codd.

En nuestro caso estudiaremos cómo aplicar hasta la Tercera Forma Normal

Tipo de dependencias.

Desarrollaremos aquí los conceptos sobre los que se basa el análisis de dependencias entre los atributos de una misma tabla, que se lleva a cabo en el proceso de normalización antes indicado, son los siguientes:

- Dependencia Funcional: Dados los subconjuntos de atributos A y B, se dice que B depende funcionalmente de A, sí, y sólo sí, para un valor de sólo puede existir un valor para B. La dependencia funcional siempre se establece entre atributos de una misma mesa. El subconjunto A se denomina determinante, puesto que A determina el valor de B. Para representar ésta dependencia funcional utilizamos la siguiente notación: $A \rightarrow B$.

Por ejemplo, sobre la relación ALUMNO se podría, entre otros, describir la siguiente dependencia funcional:

ALUMNO(DNI, Nombre, Apellido1, Apellido2, Dirección, CPostal, Localidad)

$CP:\{DNI\}$

$DNI, Nombre \rightarrow Apellidos1, Apellidos2$

- Dependencia funcional total: Dados los atributos A_1, A_2, \dots, A_k y B, se dice que B depende funcionalmente de forma completa de A_1, A_2, \dots, A_k , si y sólo si B depende funcionalmente del conjunto de atributos A_1, A_2, \dots, A_k , pero no de ninguno de sus posibles subconjuntos. Es decir, del conjunto A no se puede eliminar ningún atributo y B está formado por un único atributo.

La dependencia funcional anterior no es completa porque podemos eliminar Nombre del determinante y la implicación se seguiría cumpliendo. Por otro lado, B debe ser un solo atributo. Serían

Dependencias Funcionales Completas:

$DNI \rightarrow Apellido1$

$DNI \rightarrow Cognom2$

- Dependencia Transitiva: Dados tres atributos A, B y C, se dice que existe una dependencia transitiva entre A y C, si B depende funcionalmente de A y C depende funcionalmente de B. A, B y C podrían ser un solo atributo o un conjunto de ellos.

Per exemple en la taula anterior:

$DNI \rightarrow CPostal$

$DNI \rightarrow Localidad$

$CPostal \rightarrow Localidad$

- Conjunto de Interés: Conjunto de todas las Dependencias Funcionales Completas entre los atributos de una mesa.

Para Normalizar una mesa hasta la 3ªFN un dato del problema será el Conjunto de Interés.

Por ejemplo, dada la tabla anterior, el Conjunto de Interés sería:

ALUMNO(DNI, Nombre, Apellido1, Apellido2, Dirección, CPostal, Localidad)

CP:{DNI}

Conjunto de Interés de la mesa ALUMNO:

$DÍAS \rightarrow Nom$

$DNI \rightarrow Apellido1$

$DÍAS \rightarrow Cognom2$

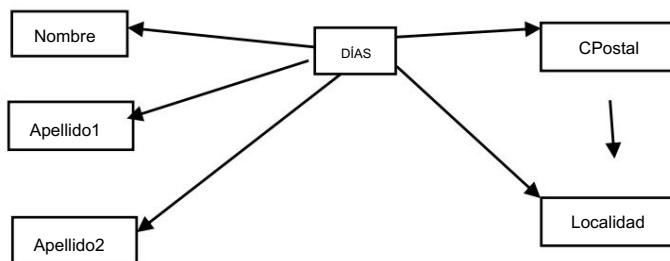
$DÍAS \rightarrow Adreça$

$DÍAS \rightarrow CPostal$

$DÍAS \rightarrow Localitat$

$CPostal \rightarrow Localidad$

Ésta es una manera de representarlo. Utilizaremos otra notación gráfica que resulta más comprensible. Así, el Conjunto de Interés lo representaremos así:



Formas Normales.

Una vez conocidos los conceptos sobre los que se basa el proceso de normalización, deben llevar a cabo una serie de etapas consecutivas en las que se aplicarán las propiedades de cada una de las formas normales definidas por Codd. A continuación se exponen los requisitos a cumplir por las tablas de nuestra base de datos según la forma normal que apliquemos.

1ra forma normal

Una tabla está en Primera Forma Normal (1FN) sí, y sólo sí, todos los atributos de la misma contienen valores atómicos. Es decir, los dominios de todos los atributos deben ser indivisibles.

El valor más pequeño al que se puede acceder con herramientas de BBDD está en el valor de un campo de una fila.

Por tanto, éste no debe ser a su vez divisible.

Consideraremos 2 casos de dominios no atómicos: dominios estructurados y dominios multivaluados.

Dominios Estructurados:

Dominios que presentan una estructura como en apartados. Por ejemplo:

ALUMNO(DNI, Nombre, Apellido1, Apellido2, Dirección{Calle, Núm., CPostal})
CP:{DNI}

En esta mesa el campo Dirección está estructurado en Calle, Nº. y CPostal. Si queremos acceder a cada uno de ellos individualmente, no es un diseño correcto incluirlos en un solo campo. La solución es muy sencilla: se deshace la estructura. Así, en lugar de un campo Dirección, tendremos:

ALUMNO(DNI, Nombre, Apellido1, Apellido2, Calle, Núm., CPostal)
CP:{DNI}

Sustituimos el campo Dirección por los campos necesarios para representar a cada uno de sus apartados.

Propietarios de multimoneda

Dominios que contienen una lista de valores de un determinado tipo. La lista tiene un tamaño indeterminada. Por ejemplo, sea la tabla:

ALUMNO(DNI, Nombre, Curso, FechaMatrícula, Tutor, Localidad, Provincia, {Teléfonos}n)

CP:{DNI}

{Teléfonos}n es un atributo multivaluado

. Veamos un posible estado de la mesa:

DÍAS	Nombre	Curs	Cerrar Registro de Tutor	Ubicación	Provincia de Alumni	Teléfonos de Alumni	
11111111A	Eva	1ESO-A 01	Julio-2016	Isabel	Écija	Sevilla	660111222
22222222B	Ana	1ESO-A 09	Julio-2016	Isabel	Écija	Sevilla	660222333 660333444 660444555
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla	
44444444D	Juan	2ESO-A 05	Julio-2016	Federico	El Villar	Córdoba	
55555555E	José	2ESO-A 02	Julio-2016	Federico	El Villar	Córdoba	661000111 661000222

Como puede observarse, esta tabla no está en 1FN ya que el campo Teléfonos contiene varios datos dentro de una misma celda y por tanto no es un campo cuyo valor sea atómico. La solución sería eliminar el campo de la tabla y crear una tabla para albergar adecuadamente los datos del campo multivaluado sin perder información; las tablas están relacionadas y sigue siendo posible acceder a los teléfonos de un determinado alumno:

ALUMNO (DNI, Nombre, Curso, Fecha Matrícula, Tutor, Localidad, Provincia)

CP:{DNI}

TELÉFONOS (DNI, Teléfono)

CP: {DNI, Teléfono}

CAj: {DNI}->ALUMNE

Así, pasaríamos a dos tablas en el siguiente estado:

ALUMNOS

DÍAS	Nombre	Curso	Cerrar Registro de Tutor	Localidad	Alumno	Provincia	Alumno
11111111A	Eva	1ESO-A 01	Julio-2016	Isabel	Écija	Sevilla	
22222222B	Ana	1ESO-A 09	Julio-2016	Isabel	Écija	Sevilla	
33333333C	Susana	1ESO-B	11-Julio-2016	Roberto	Écija	Sevilla	
44444444D	Juan	2ESO-A 05	Julio-2016	Federico	El Villar	Córdoba	
55555555E	José	2ESO-A 02	Julio-2016	Federico	El Villar	Córdoba	

TELÉFONOS

DÍAS	Teléfono
11111111A	660111222
22222222B	660222333
22222222B	660333444
22222222B	660444555
55555555E	661000111
55555555E	661000222

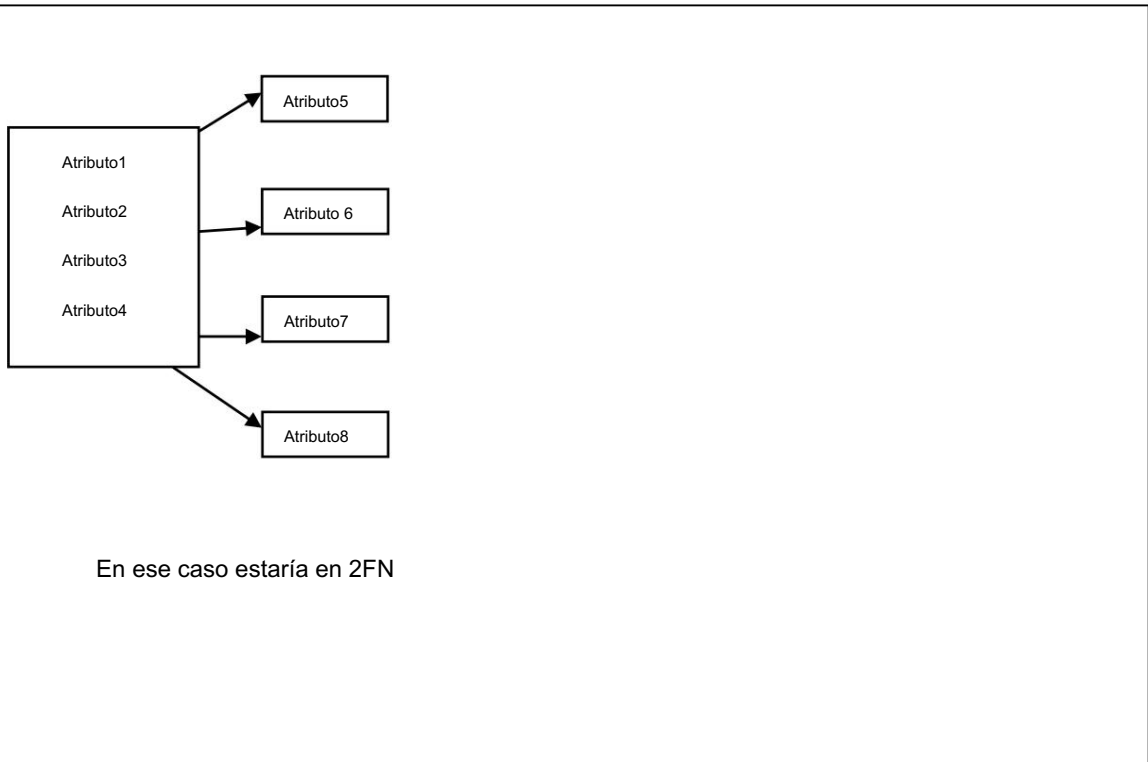
2a Formato normal

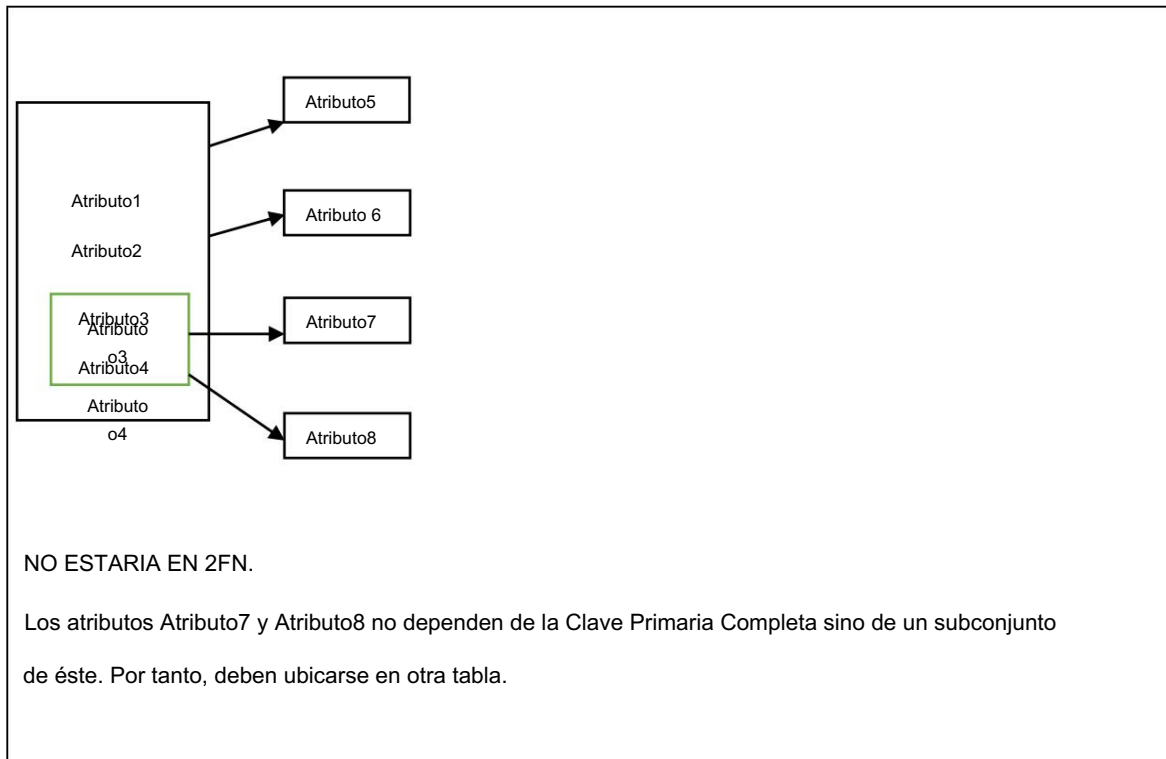
Una tabla está en Segunda Forma Normal (2FN) sí, y sólo sí, está en 1FN y, además, todos los atributos que no pertenecen a laC llave Primaria dependen funcionalmente de forma completa d'ella.

Es obvio que una tabla que esté en 1FN y cuya clave esté compuesta por un único atributo, estará en 2FN.

Es decir, se busca que los atributos de una mesa, con Clave Primaria formada por más de un atributo tengan el siguiente modelo de Conjunto de Interés:

Imaginemos una tabla con 8 atributos de los que 4 formar parte de la Clave Primaria:





Cuando una tabla está en 1FN pero no en 2FN, es necesario situar los atributos que lo impiden en las tablas adecuadas. Ésta será la que tengan como Clave Primaria el subconjunto de la Clave Primaria del que depende Funcionalmente de Manera Completa el atributo en cuestión. Pueden suceder dos cosas:

- Que la mesa exista. Y en ese caso nos limitaremos a trasladar el atributo. Si hay una Clave ajena definida sobre el subconjunto de atributos que son el DETERMINANTE del atributo, significa que la mesa existe.
- Que la mesa no exista. Y en este caso deberemos crearla y establecer la Clave Aliena pertinente en la mesa que estamos normalizando.

Ejemplo:

FACTURA(Núm., Cliente)

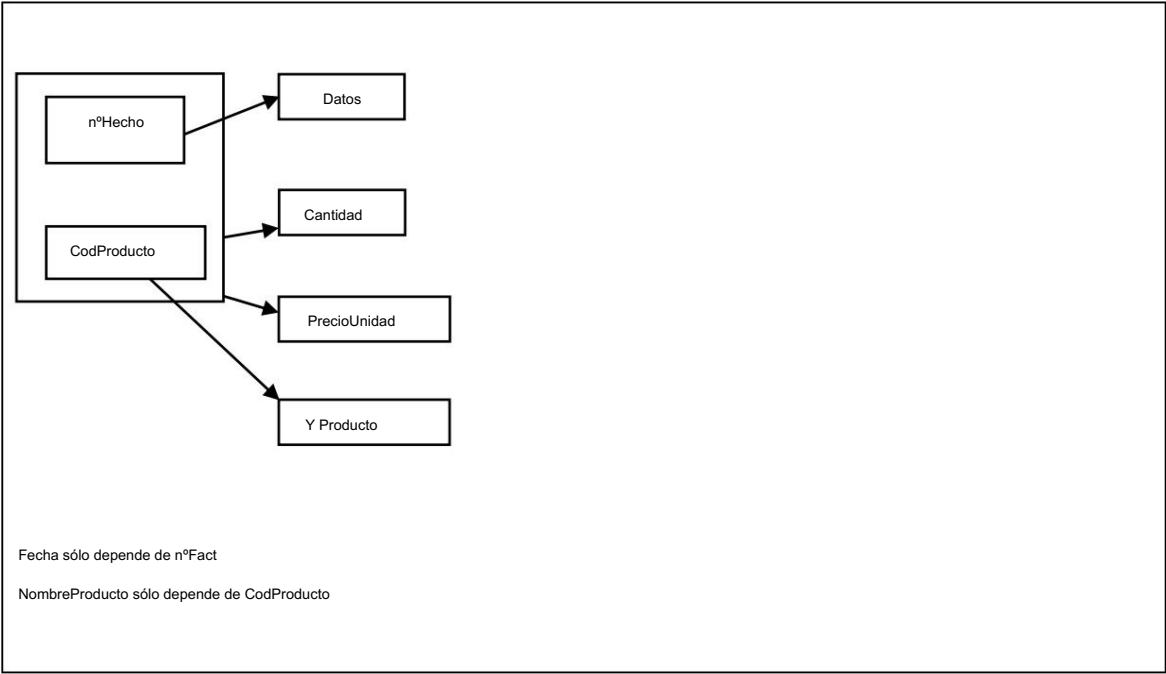
CP:{ Perilla. }

LÍNEAFACT(CodProducto,,nºFact, Fecha, Cantidad, PrecioUnidad, NombreProducto)

CP:{ CodProducto, nºFact }

CAj:{ nºFact }->FACTURA

Conjunto de Interés LÍNEAFACT:



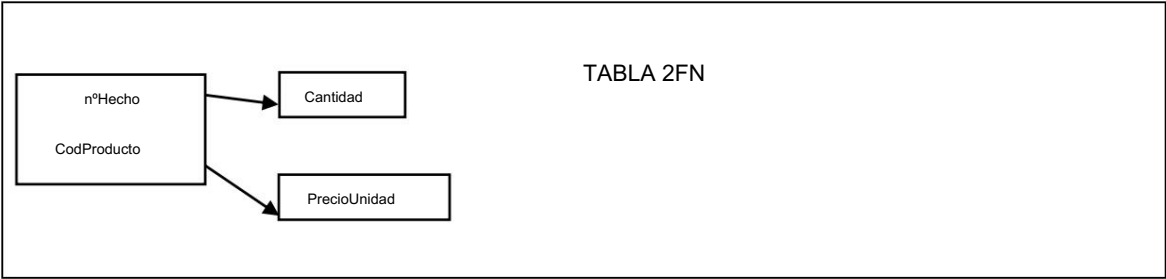
Hem de reubicar Data i NomProducto.

En el caso de Fecha, tenemos una tabla que tiene como CP el DETERMINANTE de Fecha. La Llave Ajena de la tabla LÍNEAFACT nos indica que esos valores son CP en la tabla FACTURA. Por tanto, sólo deberemos trasladar Fecha a la mesa FACTURA.

En el caso de NomProducto no existe una tabla en la que podamos situarlo por lo que habrá que crear la tabla.

FACTURA(Núm., Cliente, Fecha)	LÍNEAFACT(CodProducto,nºFact,Cantidad,
CP:{ Perilla. }	PrecioUnidad)
PRODUCTO(Cod, NombreProducto)	CP:{ CodProducto, nºFact }
CP:{ Código }	CAj:{ nºFact }->FACTURA
	CAj:{ CodProducto }->PRODUCTO

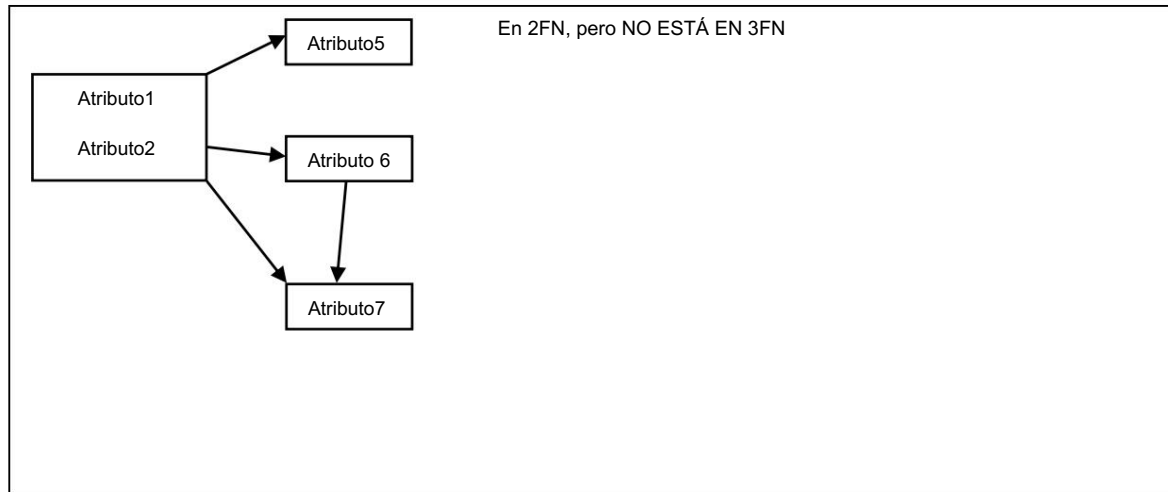
Conjunto de Interés LÍNEAFACT:



3a Forma Normal

Una tabla está en Tercera Forma Normal (3FN o FN3) sí, y sólo sí, está en 2FN y, además, cada atributo que no forma parte de la clave primaria no depende transitivamente de la clave primaria.

Es decir, NO ESTÁ EN 3FN si está en 2FN y el Conjunto de interés tiene transitividades:



La solución es reubicar los atributos que impiden que la mesa esté en 3FN en tablas que tengan como CP el Determinante que no es la CP. Así en el ejemplo de arriba, debería trasladarse Atributo7 en una mesa que tuviera como CP Atributo6.

Como en el caso anterior pueden suceder dos cosas:

- Que la mesa exista. Y en ese caso nos limitaremos a trasladar el atributo. Si hay una Clave ajena definida sobre el DETERMINANTE del atributo, significa que la mesa existe.
- Que la mesa no exista. Y en este caso deberemos crearla y establecer la Clave Aliena pertinente en la mesa que estamos normalizando.

Ejemplo:

CURSO(Código, Tutor)

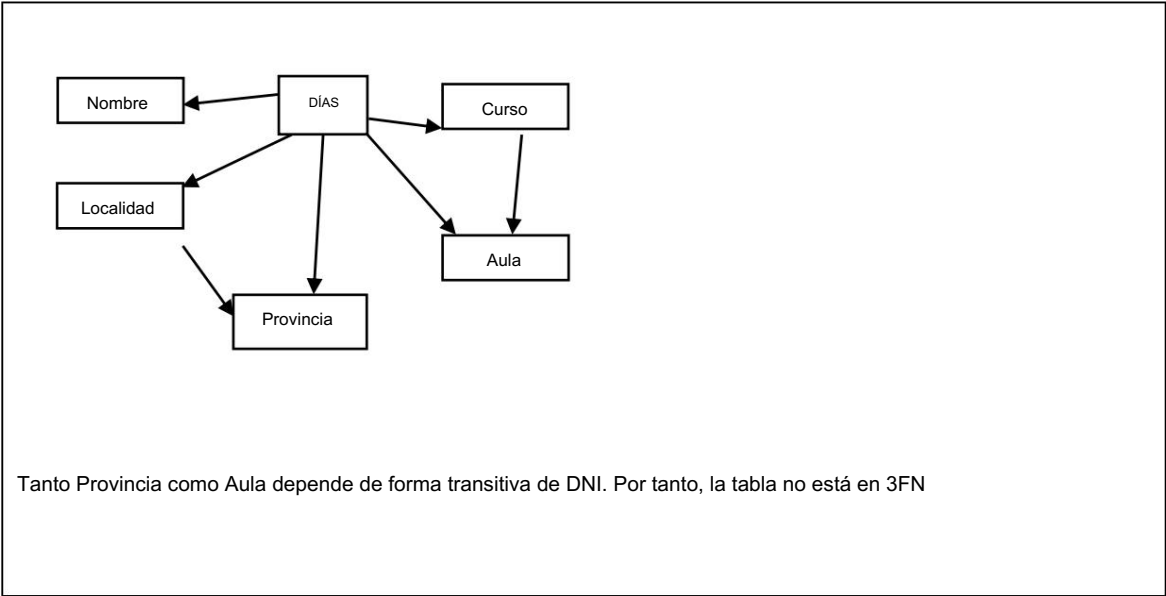
CP{Código}

ALUMNO(DNI, Nombre, Curso, Aula, Localidad, Provincia)

CP:{DNI}

CAj:{Curso}->CURSO

Conjunto de Interés de ALUMNO



En el caso de Provincia, no existe una tabla que tenga como CP Localidad por lo que habrá que crearla y conectarla con la mesa ALUMNO.

En el caso de Aula, sí existe porque los valores del atributo Curso son CP en la tabla CURSO.

SOLUCIÓN

CURSO (Código, Tutor, Aula)

CP{Código}

ALUMNO(DNI, Nombre, Curso, Localidad)

CP{DNI}

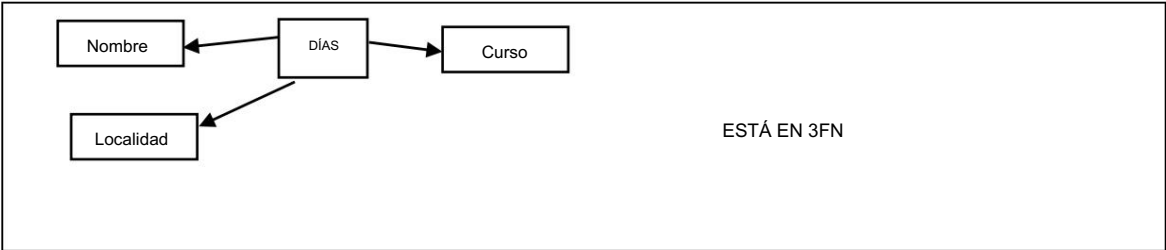
CAj{Curso}->CURSO

CAj{Localidad}->LOCALIDAD

LOCALIDAD(Cod, Provincia)

CP{Código}

Conjunto de Interés de ALUMNO



Por otra parte, las transitividades producidas porque hay más de una clave en la relación no se consideren.

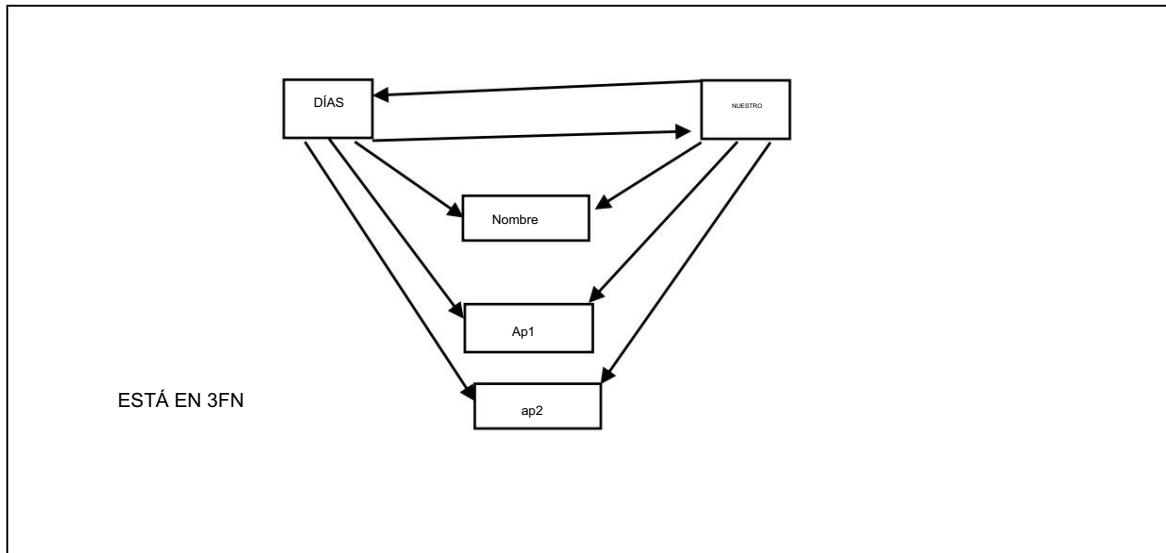
Una mesa estará en 3FN, aunque tenga transitividades, si las transitividades están producidas por la existencia de otra clave distinta a la CP. Por ejemplo:

ALUMNE(DNI, NIA, Nombre, Ap1, Ap2)

CP:{DNI}

UNI:{NIA}

Conjunto de Interés de ALUMNO



La transitividad producida por la existencia de otra clave no produce redundancia.